



Preprocesadores

Adrián Nataniel Vega Pérez 2º DAW

1. Introducción.....	2
2. SCSS.....	2
3. Conclusión.....	9
4. Webgrafía.....	10

1.Introducción

Esta es la introducción a la memoria del ejercicio evaluable llamado “Preprocesadores”. En ella se expone cómo se hizo el Scss de la página web y el porqué de las decisiones tomadas con respecto a la personalización de esta.

2.SCSS

```
// Definimos las variables para los colores de la pagina
$color: #eee;
$color_hover: lighten($color, 10%);
$color_fondo: #15a;
$color_fondo_hover: lighten($color_fondo, 10%);
$color_titulo: darken($color_fondo, 10%);
```

Empezamos definiendo todas las variables que vamos a usar para la página, estas están orientadas a la paleta de colores.

```
// Creamos un mixin para las transiciones
@mixin transition($property, $duration, $timing: linear) {
  -webkit-transition: $property $duration $timing;
  -moz-transition: $property $duration $timing;
  -o-transition: $property $duration $timing;
  transition: $property $duration $timing;
}

// Creamos un mixin para las animaciones
@mixin animation($name, $duration, $timing: linear) {
  -webkit-animation: $name $duration $timing;
  -moz-animation: $name $duration $timing;
  -o-animation: $name $duration $timing;
  animation: $name $duration $timing;
}
```

Aquí llamamos a los diferentes mixins para que los diferentes navegadores se hagan cargo de las animaciones y transiciones.

```
// Definimos la animación para los menús desplegables
@keyframes menu_desplegable {
  // Ya que display none y block es difícil de animar
  // Le quitamos y damos opacidad para darle el efecto
  from {
    opacity: 0;
  }
  to {
    opacity: 1;
  }
}
```

Definimos el keyframe para los menús desplegables, opté por usar opacity debido a que los display none y block no se pueden animar, por lo que le damos un efecto de aparición sencillo.

```
// Creamos un mixin para gestionar el responsive
@mixin responsive($size) {
  // Configuramos los estilos para pantallas medianas y pequeñas
  @if $size = medium {
    @media (max-width: 1024px) {
      @content
    }
  }
  @else if $size = small {
    @media (max-width: 800px) {
      @content
    }
  }
}
```

Creamos el último mixin del scss que se encargará de los eventos responsive de la página. Definimos tanto para pantallas medianas como pequeñas.

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}
```

Creamos los elementos generales de la página, entre ellos está box-sizing el cual hará que todos los elementos tengan el mismo “tamaño” y así que no aparezcan desalineados, como por ejemplo los input y el button.

```
// Editamos el título  
header {  
  font-size: 30px;  
  font-weight: bold;  
  background-color: $color;  
  padding: 15px 20px;  
  color: lighten(#000, 30%);  
}
```

Le damos personalización al header apoyándonos de una variable anteriormente creada.

```
// Configuramos la barra de navegación con ayuda de las variables
nav {
  background-color: $color_fondo;
  padding: 5px 0;

  ul {
    list-style-type: none;
    display: flex;
    padding: 0;

    li {
      padding: 0 70px;

      // Aplicamos una transición al fondo
      @include transition(background-color, 0.5s, ease-out);

      // Ocultamos el submenú y lo posicionamos
      ul {
        display: none;
        color: $color;
        background-color: $color_fondo;
        position: absolute;
        left: 0;
      }

      a {
        color: $color;
        text-decoration: none;

        //Le damos una transición al cambio de color
        @include transition(color, 0.5s, linear);
      }
    }
  }
}
```

Personalizamos el menú de navegación usando las variables y dándole transiciones para cuando pasamos el ratón por los diferentes enlaces, al submenú le damos display none para que por defecto esté oculto y posición absoluta para que aparezca donde queramos, por último usamos left para acomodarlo perfectamente con el menú principal.

```
// Definimos el comportamiento al pasar el ratón
&:hover {
    background-color: $color_fondo_hover;

    a {
        color: $color_hover;
    }
}

// Mostramos el submenú con la animación definida anteriormente
ul {
    display: block;

    @include animation(menu_desplegable, 0.5s, ease);

    li {
        margin: 10px 0;
    }
}

position: relative;
}
```

Definimos el hover para aplicar los colores junto a las transiciones, además usamos “@include animation” para darle el efecto antes configurado. También usamos position relative en el menú principal para alinear los submenús.

```
// Ocultamos el menú de navegación en pantallas pequeñas
@include responsive(small) {
    display: none;
}
```

Para las pantallas pequeñas configuramos que no aparezca el menú de navegación

```
// Estructuramos el contenido principal
main {
  padding-bottom: 20px;

  // Hacemos que sea float para que ocupe el lado izquierdo
  article:first-child {
    float: left;
    width: 25%;
    padding: 20px;

    form {
      p {
        margin-bottom: 15px;

        input, button {
          // Definimos que ocupen el 90% del article
          width: 90%;
          padding: 5px 0;
          border-radius: 5px;
          border: 1px solid gray;

          // Ajustamos el ancho al 100%
          @include responsive(small) {
            width: 100%;
          }
        }

        input {
          padding-left: 10px;
        }

        button {
          color: $color;
          background-color: $color_fondo;

          // Debido a que son varios efectos los que queremos aplicar
          // Pasamos como variable que use todos los efectos posibles
          @include transition(all, 0.3s, ease-out);
        }
      }
    }
  }
}
```

Personalizamos el main usando float para mandar el formulario a la izquierda y el texto a la derecha y le damos un tamaño por porcentaje a la página, hacemos lo mismo con el input y los botones para que rellenen el article y en el formato responsive rellenamos el elemento entero por los inputs y el button. Adicionalmente añadimos una transición al botón para que cuando pase el ratón se apliquen todos los efectos posibles.

```
&:hover {  
    background-color: $color_fondo_hover;  
    color: $color_hover;  
}
```

Aquí le decimos que cuando pase el ratón aplique los colores junto al efecto.

```
// Quitamos el modo float al formulario para que se apilen  
@include responsive(small) {  
    float: none;  
    width: 100%;  
    padding: 50px 20px;  
}
```

Cuando se trate de un móvil los apilamos en bloques y hacemos que ocupe toda la página.

```
// Configuramos la columna de contenido principal  
article:last-child {  
    float: left;  
    width: 70%;  
    margin: 20px 0;  
  
    h1 {  
        font-size: 20px;  
        color: $color_titulo;  
    }  
  
    p {  
        margin: 20px 0;  
    }  
  
    // Quitamos el modo float para apilar los elementos  
    @include responsive(small) {  
        float: none;  
        width: 95%;  
        margin: 0 20px;  
    }  
}
```

Añadimos float para ajustar el texto y también el comportamiento responsive.


```
footer {  
  clear: left;  
  color: $color;  
  background-color: $color_fondo;  
  padding: 5px;  
  
  // Fijamos el footer y lo posicionamos  
  position: fixed;  
  bottom: 0;  
  left: 0;  
  width: 100%;  
}
```

Usamos clear para que no sea afectado por los float, le aplicamos los colores de las variables y le damos el formato fixed para que permanezca siempre en visión, usando bottom y left para posicionarlo y width para que ocupe toda la página.

3. Conclusión

Se han usado las diferentes herramientas que proporciona el preprocesador SCSS para aplicar de forma rápida y sencilla la personalización de la página, ya sean efectos como transiciones o animaciones o formato responsive. También se ha descubierto la utilidad de las variables para evitar repetir código.

4. Webgrafía

Apuntes de clase:

[Transiciones y animaciones](#)

Float:

[W3Schools - Float](#)

Animaciones:

[Falconmasters](#)

[Vídeo Germán Rodríguez](#)